



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/643,585	08/18/2003	Steven L. Scott	1376.700US1	4004
21186	7590	09/15/2008	EXAMINER	
SCHWEGMAN, LUNDBERG & WOESSNER, P.A. P.O. BOX 2938 MINNEAPOLIS, MN 55402				TSAI, SHENG JEN
2186		ART UNIT		PAPER NUMBER
09/15/2008		MAIL DATE		DELIVERY MODE
				PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)
	10/643,585	SCOTT, STEVEN L.
	Examiner	Art Unit
	SHENG-JEN TSAI	2186

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1) Responsive to communication(s) filed on 08 July 2008.
 2a) This action is FINAL. 2b) This action is non-final.
 3) Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4) Claim(s) 1,3-8 and 11-18 is/are pending in the application.
 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
 5) Claim(s) _____ is/are allowed.
 6) Claim(s) 1,3-8 and 11-18 is/are rejected.
 7) Claim(s) _____ is/are objected to.
 8) Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9) The specification is objected to by the Examiner.
 10) The drawing(s) filed on 08 July 2008 is/are: a) accepted or b) objected to by the Examiner.
 Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
 Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
 11) The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12) Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
 a) All b) Some * c) None of:
 1. Certified copies of the priority documents have been received.
 2. Certified copies of the priority documents have been received in Application No. _____.
 3. Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) <input type="checkbox"/> Notice of References Cited (PTO-892)	4) <input type="checkbox"/> Interview Summary (PTO-413)
2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948)	Paper No(s)/Mail Date. _____ .
3) <input checked="" type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date <u>7/8/2008</u> .	5) <input type="checkbox"/> Notice of Informal Patent Application
	6) <input type="checkbox"/> Other: _____ .

DETAILED ACTION

1. This Office Action is taken in response to Applicants' Request for Continued Examination (RCE) filed on 07/08/2008 regarding Application 10/643,585 filed on 08/18/2003.

Claim Objections

2. Claim 18 is objected to because of the following informalities:

Claim 18 as filed on 4/23/2007 recites "The method of claim 18, wherein ..." Since a claim cannot depend from itself, claim 18 is an improper dependent claim. It appears that claim 18 is intended to depend from claim 17 instead of claim 18.

Appropriate correction is required.

Claim Rejections - 35 USC § 103

3. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

4. Claims 1, 3, 5-8 and 11-18 are rejected under 35 U.S.C. 103(a) as being unpatentable over Scott et al. (US 6,925,547, hereinafter referred to as Scott), and in view of Fossum et al. (US 4,888,679; hereinafter referred to as Fossum).

As to claim 1, Scott discloses **a computer system** [as shown in figures 1-3] comprising:

a network [interconnection network, figure 2, 14],

one or more processing nodes connected via the network [as shown in figures 1-3; figure 2 shows nodes A and B, with node A having processors 24A and 26A while node B having processors 24B and 26B], **wherein each processing node includes: a scalar processing unit, a vector processing unit and means for operating the scalar processing unit independently of the vector processing unit** [taught by Fossum et al., see below], **a plurality of processors** [PM, figure 1, 12; figure 2 shows nodes A and B, with node A having processors 24A and 26A while node B having processors 24B and 26B], **a processor cache** [column 5, lines 48-53; To support local address translations, each SHUB contains a translation-lookaside buffer (TLB) 108 for performing local address translations for both block transfers and AMOs. A TLB is a cache that holds only page table mappings (column 16, lines 7-15)] **and a translation look aside buffer (TLB)** [abstract; column 1, lines 40-53; To support local address translations, each SHUB contains a translation-lookaside buffer (TLB) 108 for performing local address translations for both block transfers and AMOs. A TLB is a cache that holds only page table mappings (column 16, lines 7-15); Section "local Address Translation" describes in details how local address translation is done by using a LCT (Local Connection Table) (col. 14, line 54 to col. 17 line 30) and an "external TLB" located on the local SHUB (col. 14, lines 47-52)], **wherein the scalar processing unit places instructions for the vector processing unit in a queue for execution by the vector processing unit and the scalar processing unit continues to execute additional instructions** [taught by Fossum et al., see below]; and

a shared memory, wherein the shared memory is connected to each of the processors within the processing node [memory, figure 2, 28A and 28B, where memory 28A is shared by processors 24A and 26A and memory 28B is shared by processors 24B and 26B; column 5, lines 47-67], **wherein the shared memory includes a Remote Address Translation table (RTT)** [The SHUB at each node of multiprocessor system 10 contains an external TLB to perform address translations for both block transfers and AMOs ... (col. 17, lines 33-46); Section “Remote Address Translation” (col. 17 line 31 to col. 20, line 57) provides description in details regarding how remote address translation is done], **wherein the RTT contains translation information for an entire virtual memory address space** [column 17, lines 35-45; column 2, lines 65-67 and column 3, lines 1-23; note that the RTT contains translation information for a virtual memory space of at least the particular remote node in order to be able to perform address translation for requests from other nodes] **wherein the RTT translates memory addresses received from other processing node such that the memory addresses are translated into physical addresses within the shared memory** [Section “Remote Address Translation” (col. 17 line 31 to col. 20, line 57) provides description in details regarding how remote address translation is done; A method of performing remote address translation in a multiprocessor system includes determining a connection descriptor and a virtual address at a local node, accessing a local connection table at the local node using the connection descriptor to produce a system node identifier for a remote node and a remote address space number, communicating the virtual address and remote address space number to the remote

node, and translating the virtual address to a physical address at the remote node (qualified by the remote address space number) (abstract); figures 4A, 4B, 5A and 5B; column 25, lines 39-50];

wherein processors on one node can load data directly from and store data directly to shared memory on another processing node via addresses that are translated on the other processing node using the other processing node's RTT

[In such a system, each processor can directly access all of memory, including its own local memory and the memory of the other (remote) processing element nodes .. (col. 1, lines 38-53); The SHUB at each node of multiprocessor system 10 contains an external TLB to perform address translations for both block transfers and AMOs ...

(col. 17, lines 33-46); Section "Remote Address Translation" (col. 17 line 31 to col. 20, line 57) provides description in details regarding how remote address translation is done; abstract; figures 4A, 4B, 5A and 5B; column 25, lines 39-50]; **and**

wherein each TLB exists separate from the RTT [The address translation mechanism used by CE 64 uses an external TLB located on the local SHUB, or an external TLB located on a remote SHUB (col. 14, lines 47-49); thus the corresponding TLB that performs "local address translation" is located on the local SHUB and the corresponding RTT that performs "remote address translation" is located on the remote SHUB, and therefore the TLB and RTT are separate because one is located at the local SHUB and the other is located at the remote SHUB] **and wherein each TLB translates memory references from its associated processor to the shared memory within the processing node** [Section "local Address Translation" describes

in details how local address translation is done by using a LCT (Local Connection Table) (col. 14, line 54 to col. 17 line 30) and an “external TLB” located on the local SHUB (col. 14, lines 47-52); As described above, the local TLB can be used by a local CE 64 to perform translations for local memory accesses, thereby allowing the user to program the CE using virtual addresses. As now described, CE 64 can also be programmed to send virtual addresses to a remote or target node for remote memory accesses (using the CD associated with the virtual address to identify the remote node), with the TLB on that node being used to translate those addresses (column 17, lines 35-45); see also column 2, lines 65-67 and column 3, lines 1-23); Section "local Address Translation" describes in details how local address translation is done by using a LCT (Local Connection Table) (col. 14, line 54 to col. 17 line 30) and an “external TLB” located on the local SHUB (col. 14, lines 47-52)].

Regarding claim 1, Scott does not teach that **each processor includes a scalar processing unit, a vector processing unit and means for operating the scalar processing unit independently of the vector processing unit.**

However, the concepts of scalar processors and vector processors is well known and widely used in the art. Essentially every PC has a scalar processor for data processing, and vector processors are commonly used for graphic applications (see Microsoft Computer Dictionary, 5th edition, 2002, Microsoft Press, page 548 – vector and page 549 – vector graphics).

Further, Fossum discloses in their invention “Method and Apparatus Using a Cache and Main memory for Both Vector Processing and Scalar Processing by

Prefetching Cache Blocks Including Vector Data Elements" an apparatus comprising a vector processor (figure 1, 22; figure 7, 116) and a scalar processor (figure 1, 21; figure 7, 108) where the scalar processor and the vector processor operate independently of each other (figure 7; column 2, lines 35-68; column 3, lines 1-43). Including both scalar and vector processors in a computer system with a cache allows the prefetching of block data using the vector processor and increases the data throughput (column 2, lines 12-34).

Specifically, Fossum discloses that **each processor includes a scalar processing unit, a vector processing unit and means for operating the scalar processing unit independently of the vector processing unit** [a vector processor (figure 1, 22) is added to a digital computing system 9figure 1, 20) including a scalar processor (figure 1, 21), a virtual address translation buffer, a main memory (figure 1, 23), and a cache (figure 1, 24) (column 3, lines 7-10); figure 7 shows the detailed organization of these components], **wherein the scalar processing unit places instructions for the vector processing unit in a queue for execution by the vector processing unit** [Another object of the invention is to take a main memory and cache optimized for scalar processing and make it suitable for vector processing as well (column 2, lines 40-42); in accordance with the invention, a main memory and cache suitable for scalar processing are used in connection with a vector processor by issuing prefetch requests in response to the recognition of a vector load instruction (column 2, lines 47-51); In response to a vector load instruction, the scalar processor executes microcode for sending a vector load command to the vector processor, and also for

sending the vector prefetch requests to the cache. The vector prefetch requests include the virtual addresses of the blocks that will be accessed by the vector processor. These virtual addresses are computed based upon the vector address, the length of the vector, and the stride or spacing between the addresses of the adjacent elements of the vector (column 3, lines 17-26); FIG. 7 is a preferred embodiment of the present invention which uses microcode in a scalar processing unit to generate vector prefetch requests for an associated vector processing unit (column 3, lines 67-68); **column 11, lines 35-46] and the scalar processing unit continues to execute additional instructions** [Specifically, the scalar processing unit includes a micro-sequencer and issue logic 109 which executes prestored microcode 110 to interpret and execute the parsed instructions from the instruction processing unit 107. These instructions include scalar instructions which the micro-sequencer and issue logic executes by operating a register file and an arithmetic logic unit 111. These scalar instructions include, for example, an instruction to fetch scalar data from the cache unit 106 and load the data in the register file 111 (column 11, lines 35-46)].

It is well known in the art that the use of vector processors increases the throughput by processing multiple vector elements simultaneously as opposed to processing a single element at a time.

Therefore, it would have been obvious for one of ordinary skills in the art at the time of Applicant's invention to recognize the benefit of having both scalar and vector processing units, as demonstrated by Fossum, and to incorporate it into the existing apparatus disclosed by Scott to further enhance the performance of the system.

As to claim 3, Scott teaches that **the shared memory further includes a plurality of cache coherence directories, wherein each processing node is coupled to one of the cache coherence directories** [In one embodiment, all of the coherence information is passed across the bus in the form of messages, and each processor on the bus "snoops" by monitoring the addresses on the bus and, if it finds the address of data within its own cache, invalidating that cache entry. Other cache coherence schemes can be used as well (column 5, lines 47-67)].

As to claim 5, Scott teaches that **the processing nodes include at least one input/out (I/O) channel controller** [I/O, figure 1, 18], **wherein each I/O channel controller is coupled to the shared memory of the processing node** [figures 1-3; column 4, lines 10-22].

As to claim 6, Fossum teaches that **each scalar processing unit contains a scalar cache memory** [cache, figure 1, 24 is associated and shared by the scalar (21) and vector (22) processing units], **wherein scalar cache memory contains a subset of cache lines stored in the shared memory cache** [column 4, lines 15-54]; **a plurality of address latches each of which for outputting register set address bit by latching a address, in response to the register set control signal and the self-refresh signal when the mode register set signal is applied** [column 8, lines 3-18]; **and**
a partial array self-refresh controller for selectively activating the plurality of control signals by decoding the plurality of register set addresses depending on

input of the internal address [the refresh controller, figure 2, 217; column 6, lines 39-45].

As to claim 7, Scott teaches that **the network includes a router connecting one or more of the processing nodes** [R (Router), figure 1, 16]

As to claim 8, it recites substantially the same limitations as in claim 1, and is rejected for the same reasons set forth in the analysis of claim 1. Refer to “As to claim 1” presented earlier in this Office Action for details.

As to claim 11, it recites substantially the same limitations as in claim 3, and is rejected for the same reasons set forth in the analysis of claim 3. Refer to “As to claim 3” presented earlier in this Office Action for details.

As to claim 12, it recites substantially the same limitations as in claim 1, and is rejected for the same reasons set forth in the analysis of claim 1. Refer to “As to claim 1” presented earlier in this Office Action for details.

As to claim 13, it recites substantially the same limitations as in claim 3, and is rejected for the same reasons set forth in the analysis of claim 3. Refer to “As to claim 3” presented earlier in this Office Action for details.

As to claim 14, it recites substantially the same limitations as in claim 5, and is rejected for the same reasons set forth in the analysis of claim 5. Refer to “As to claim 5” presented earlier in this Office Action for details.

As to claim 15, it recites substantially the same limitations as in claim 6, and is rejected for the same reasons set forth in the analysis of claim 6. Refer to “As to claim 6” presented earlier in this Office Action for details.

As to claim 16, it recites substantially the same limitations as in claim 7, and is rejected for the same reasons set forth in the analysis of claim 7. Refer to “As to claim 7” presented earlier in this Office Action for details.

As to claim 17, it recites substantially the same limitations as in claim 1, and is rejected for the same reasons set forth in the analysis of claim 1. Refer to “As to claim 1” presented earlier in this Office Action for details.

As to claim 18, it recites substantially the same limitations as in claim 3, and is rejected for the same reasons set forth in the analysis of claim 3. Refer to “As to claim 3” presented earlier in this Office Action for details.

5. Claim 4 is rejected under 35 U.S.C. 103(a) as being unpatentable over Scott et al. (US 6,925,547, hereinafter referred to as Scott), in view of Fossum et al. (US 4,888,679, hereinafter referred to as Fossum), and further in view of Nakazato (US 6,782,468).

As to claim 4, Scott in view of Fossum does not teach that **each processor includes two vector pipelines.**

However, Nakazato discloses in the invention “Shared Memory Type Vector Processing System, Including a Bus for Transferring a Vector Processing Instruction, and Control Method Thereof” an apparatus comprising multiple vector pipelines in each processor (n vector processing units, figure 2, 14a~14n) and a scalar processor (figure 2, 11). Including multiple vector processors in a computer system allows the multiple vector processing tasks to be performed simultaneously and increases the data throughput.

Therefore, it would have been obvious for one of ordinary skill in the art at the time of Applicant's invention to recognize the benefit of having multiple vector processing units, as demonstrated by Nakazato, and to incorporate it into the existing apparatus disclosed by Scott in view of Fossum to further enhance the performance of the system.

6. *Related Prior Art*

The following list of prior art is considered to be pertinent to applicant's invention, but not relied upon for claim analysis conducted above.

- Schimmel, (US 6,105,113), "System and Method for Maintaining Translation Look-Aside Buffer (TLB) Consistency."
- Scott, (US 6,922,766), "Remote Translation Mechanism for a Multi-Node System."
- Nesheim et al., (US 5,897,664), "Multiprocessor System Having Mapping Table in Each Node to Map Global Physical Addresses to Local Physical Addresses of Page Copies."
- Vishin et al., (US 5,860,146), "Auxiliary Translation Lookaside Buffer for Assisting in Accessing Data in Remote Address Space."
- Deneau, (US 6,684,305), "Multiprocessor System Implementing Virtual Memory Using a Shared Memory, and a Page Replacement Method for Maintaining Paged memory Coherence."

- Frank et al., (US 6,490,671), "System for Efficiently Maintaining Translation Lookaside Buffer Consistency in a Multi-Threaded, Multi-Processor Virtual Memory System."
- Hansen, (US 6,101,590), "Virtual Memory System with Local and Global Virtual Address Translation."

Conclusion

7. Claims 1, 3-8 and 11-18 are rejected as explained above.
8. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Sheng-Jen Tsai whose telephone number is 571-272-4244. The examiner can normally be reached on 8:30 - 5:00.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Matthew Kim can be reached on 571-272-4182. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

/Sheng-Jen Tsai/

TFSA Examiner, Art Unit 2186

Application/Control Number: 10/643,585

Art Unit: 2186

August 27, 2008

Page 14